

ARL-FLIGHT-MECH-TM-452

AR-006-656

AD-A250 317



DTIC
ELECTE
MAY 22 1992
S C D

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
AERONAUTICAL RESEARCH LABORATORY
MELBOURNE, VICTORIA

Flight Mechanics Technical Memorandum 452

**DIGITAL ENHANCEMENT AND RESTORATION
OF IMAGES DERIVED FROM FLOW
VISUALIZATION EXPERIMENTS**

by

M.F. LEE

92-13553

Approved for public release

© COMMONWEALTH OF AUSTRALIA 1991

DECEMBER 1991

92 5 21 007

This work is copyright. Apart from any fair dealing for the purpose of study, research, criticism or review, as permitted under the Copyright Act, no part may be reproduced by any process without written permission. Copyright is the responsibility of the Director Publishing and Marketing, AGPS. Enquiries should be directed to the Manager, AGPS Press, Australian Government Publishing Service, GPO Box 84, CANBERRA ACT 2601.

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
AERONAUTICAL RESEARCH LABORATORY

Flight Mechanics Technical Memorandum 452

**Digital Enhancement and Restoration of Images Derived from Flow
Visualization Experiments**

by

M. F. Lee

Summary

The Pixedit program has been rewritten to enable image processing of raster image files residing on the IRIS 4D workstations. Such techniques as spectral filtering, edge detection and enhancement and false colouring techniques have been incorporated into the new version of the program. These techniques are usefully applied in enhancing the quality of photographic images obtained from flow visualization experiments. This report describes the techniques used, giving examples of their application to flow visualization and provides instructions for operating the revised Pixedit program.



(C) COMMONWEALTH OF AUSTRALIA 1991

POSTAL ADDRESS: Director, Aeronautical Research Laboratory,
506 Lorimer Street, Fishermens Bend,
Victoria, 3207, Australia



Accession For	
NTIS	ORNL
DTIC	PAR
Unpublished	
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

Contents

1. Image Processing	1
1.1 Introduction	1
1.2 Histogram Manipulation Techniques	2
1.2.1 Histogram Expansion	2
1.2.2 Histogram Equalization	3
1.2.3 False Colouring	4
1.3 Convolution Techniques	6
1.3.1 Noise Filtering	6
1.3.2 Edge Enhancing	10
1.4 Spectral Techniques	11
1.4.1 Spectral Filtering	12
1.4.2 Spectral Convolution	17
1.5 Edge Detection	17
2. The Pixedit Program	20
2.1 The Main Menu	20
2.2 The Image Menu	21
2.3 The Image In Memory	22
2.4 The <i>Paint</i> Function	23
2.5 The <i>Modify Colour</i> Function	23
2.6 The <i>Filter</i> Function	25
2.7 The <i>Fourier Transform</i> Function	25
3. Conclusion	26

Distribution List

Document Control Data

1. Image Processing

1.1 Introduction

Image processing is used to improve the visual quality of an image. Visual quality has two aspects, an aesthetic aspect and an informative aspect. The aesthetic aspect has more to do with art than science. This report deals more with enhancing the informative aspect of images. Images are very useful means of conveying a great deal of information quickly to a human observer. Image processing is necessary when information stored in an image is not readily perceivable or confused by interference of some sort. There are many factors that effect the informative quality of an image, poor contrast, low resolution, signal noise, etc. Theoretically, it is not possible to gain any additional information from an existing image by using image processing. It is however possible to make the existing information more meaningful to a human observer.

A digital image is an array (generally rectangular) of integer numbers which represent the colour intensity of the image at each picture element. The data for an image can be generated entirely digitally as is the case with computer generated graphics or it can be generated by digitizing some external analog source. The later source is generally more susceptible to degradations in image quality and generally benefit more from image processing techniques. Image processing is most commonly applied to video or photographic media where often poor lighting or camera work may cause information in a scene to be lost to the human observer in the resultant image. The visual quality of an image that has been digitized may be further degraded by the introduction of signal noise in the digitizing process. Once digitization has been achieved however there is little or no chance of further noise being introduced, which is a major advantage of digital images. The major disadvantage however is that in most cases a digital image has significantly less resolution than the original analog image, both spatially and in colour intensity. Figure 1 is an example of an image digitized from a video camera signal. This image was obtained from the Flight Mechanics water tunnel in which dye is used to visualize vortex breakdown over a model of the F/A-18 aircraft. Note that the image has had computer generated text inserted. This is another useful feature of digital images.

Generally digital images are of two basic types, gray scale and colour. A gray scale image has a single intensity value defined for each pixel (picture element) which represents a gray level between black and white. Colour images however require three intensities to be defined for each pixel, corresponding to three colour components. The most commonly used method for defining a colour is by specifying its red, green and blue intensity components. While this is not the only means of specifying a colour it is the most widely used because it models the function of a colour video monitor where red green and blue phosphors in the screen are each excited by an electron gun. Each of the intensities in an image is usually represented by an n -bit number giving 2^n shades of each colour component. 256 shades is generally a sufficient intensity

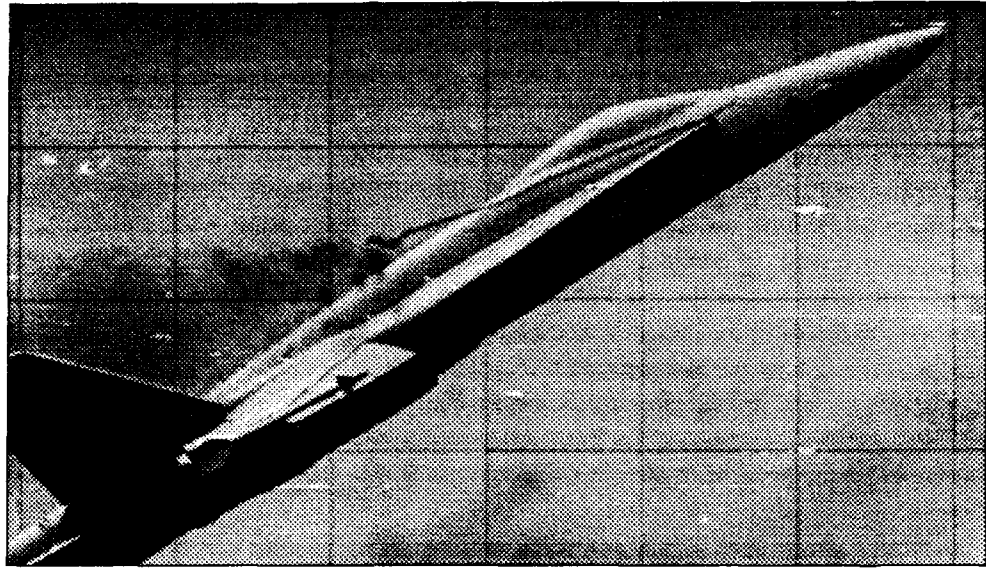


Figure 1: An example of a digital gray scale image

resolution for image applications. Thus 8-bits or one byte of storage per component per pixel is required. Colour images will thus require three bytes per pixel while gray scale images may be defined with a single byte for each pixel.

1.2 Histogram Manipulation Techniques

A simple method of improving an images appearance is to enhance the contrast of an image. The contrast of an image is loosely defined as the difference in intensity between separate segments of an image. The human eye cannot discern small detail when the contrast of the gray scales which defines that detail is also small.

Histogram manipulation techniques provide a systematic method for enhancing the contrast of an image. (ref [1]) An image histogram is a plot of the relative number of pixels of each gray scale intensity found in the image. From this it is possible to discern the intensity levels that are being under-utilized and those that are being over-utilized. An image with the maximum possible contrast will have intensities spanning the whole range from white to black. The top image in Figure 2 shows the image from Figure 1 with it's histogram. Note that this image does not have maximum contrast.

1.2.1 Histogram Expansion

Histogram expansion can be applied to an image whose range of intensities does not span the full intensity range. In most gray scale images the full intensity range spans from zero (black) to 255 (white), giving 256 shades of gray. An image's intensity range can be expanded to span this full range using a simple linear mapping of pixel

intensity. The intensity of any given pixel becomes,

$$i' = \frac{i - i_{\min}}{i_{\max} - i_{\min}} \times 255$$

where i is the pixel's intensity in the old image, i' is the pixel's intensity in the new image, and i_{\min} and i_{\max} are the minimum and maximum pixel intensities in the old image. The second image in Figure 2 shows the original image with its histogram expanded. Note the improvement in contrast between the dye downstream of the burst and the background.

1.2.2 Histogram Equalization

In some cases an image may have a band of intensities where an interesting feature is suppressed due to poor contrast. In this case histogram expansion is of little use since the contrast of the band will be only marginally improved. Histogram equalization seeks to produce an even distribution of gray scale intensities throughout the image by increasing the contrast of the more prolific intensity bands and decreasing the intensity of the less prolific intensity bands.

To see how histogram equalization is formulated it is assumed that the intensity domain is continuous instead of discrete. Then the number density of pixels at a given intensity level, $n(i)$, can be defined by a continuous function of intensity $n(i)$,

$$dN = n(i)di \quad (1)$$

where dN is the number of pixels in the image with intensities in the range $[i, i+di]$. Thus the number of pixels with intensity less than or equal to intensity i_0 is given by,

$$N(i_0) = \int_0^{i_0} n(i)di$$

Histogram equalization requires that $n(i)$ be constant for all i . This may be achieved using an intensity mapping defined by,

$$i' = \frac{i_{\max}}{N(i_{\max})} N(i)$$

where i_{\max} is the maximum available intensity in the gray scale range. Thus $N(i_{\max})$ represents the total number of pixels in the image. Since this mapping function increases monotonically, it can be shown that all pixels with intensity in the domain $i \in [i_0, i_0+di]$ and only these pixels will have new intensities in the range $i' \in [i'_0, i'_0+di']$ where,

$$di' = \frac{di'}{di} di$$

Thus the number of pixels in the original interval di and the mapped interval di' must be equal.

That is,

$$\begin{aligned} n(i)di &= n'(i')di' \\ \Rightarrow N'(i') &= N(i) \end{aligned}$$

Now from equation 1,

$$\begin{aligned} n'(i') &= \frac{dN'}{di'} \\ &= \frac{d}{di'}[N'(i')] \\ &= \frac{d}{di'}[N(i)] \\ &= \frac{dN/di}{di'/di} \\ &= \frac{1}{i_{\max}/N(i_{\max})} \frac{dN/di}{dN/di} \\ &= \frac{N(i_{\max})}{i_{\max}} \end{aligned}$$

Which is a constant as required. To apply this mapping to images with discrete intensity levels a discrete approximation of the ideal continuous mapping is used,

$$i' = \frac{i_{\max}}{N_o} \sum_{r=0}^i n_r$$

The third image in Figure 2 shows the effect of histogram equilization. Note that the histogram is *not* flat: the more frequently occurring intensities are more spread out than the less frequent intensities. In general it is not possible to achieve a perfectly uniform distribution when the domain is discrete.

1.2.3 False Colouring

Instead of using the systematic intensity mappings described above, an arbitrary function can be defined which maps gray scale to gray scale or, more usefully, three separate functions which map a gray scale intensity to each of the three colour components, red, green and blue.

$$\begin{aligned} r &= r(i) = f_1(i) \\ g &= g(i) = f_2(i) \\ b &= b(i) = f_3(i) \end{aligned}$$

thus generating an image where each gray scale intensity is mapped to an RGB (red, green, blue) colour.

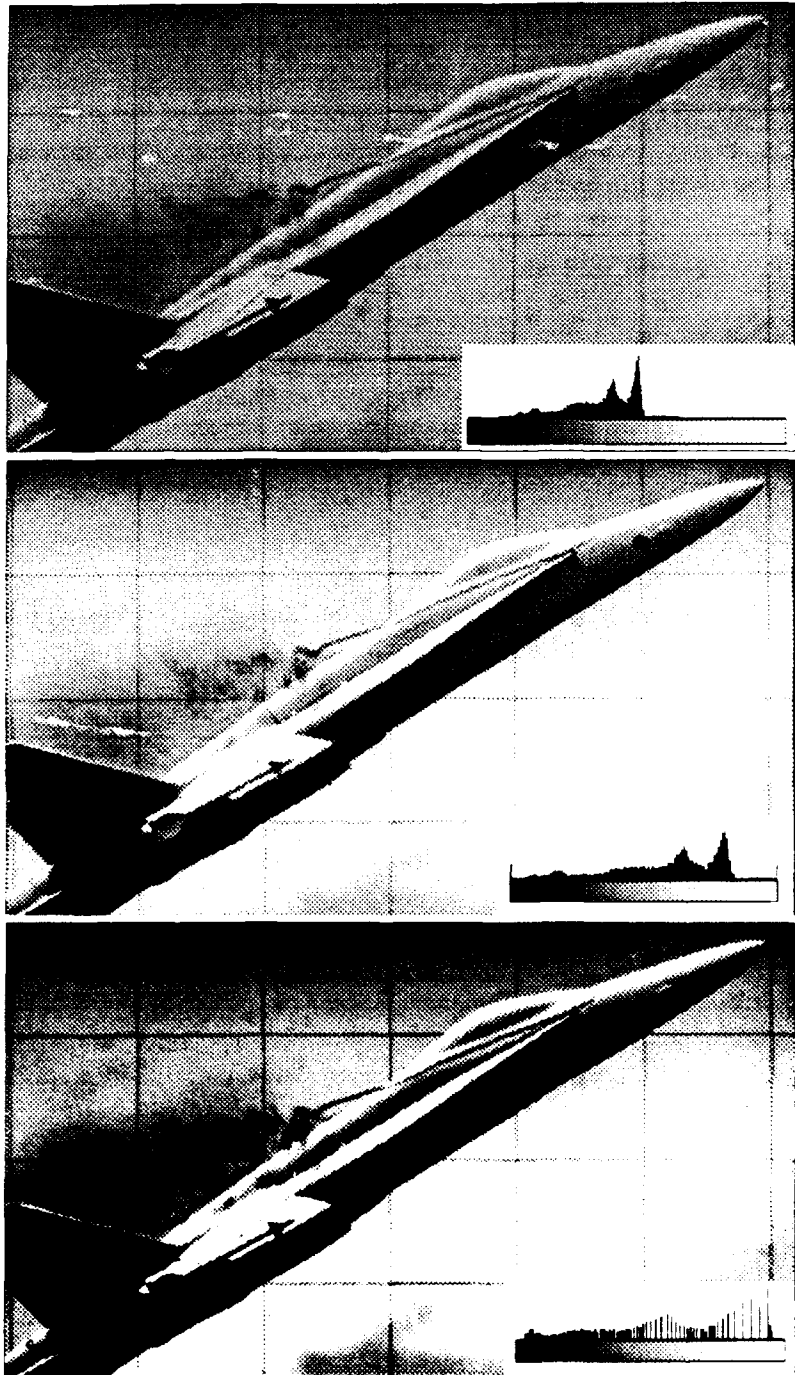


Figure 2: The top picture shows the histogram of the image from Figure 1. The second is the same image with its histogram expanded and the third shows the image after histogram equalization.

The definitions of the mapping functions may be arbitrarily chosen to accentuate relevant features in the image, or to hide some unwanted noise aspect. The transformation could possibly be selectively applied to certain regions of the image so that a particular feature is coloured and stands out from the rest of the image which could be left gray. Figure 3 shows a magnified image of the vortex burst region where only the dye in the image has had false colour added.

1.3 Convolution Techniques

By examining a pixel in the context of its nearest neighbours it is possible to gain information about the image at that point. For instance if the pixel forms part of an edge or a line, or if the pixel has been corrupted by noise. Convolution processes are a simple way of enhancing such information in an image. The convolution of one function, $f(x,y)$, with another, $g(\alpha, \beta)$, is given by:

$$[f * g](x, y) = \iint g(\alpha, \beta) f(x - \alpha, y - \beta) d\alpha d\beta$$

If $f(x, y)$ is considered as the image definition and $g(\alpha, \beta)$ to be some convolving function or mask, then in the discrete domain of an image this definition translates to:

$$[f * g]_{x,y} = \sum_{\beta=0}^{n-1} \sum_{\alpha=0}^{m-1} g_{\alpha,\beta} f_{x-\alpha, y-\beta}$$

In most practical applications $m = n = 3$ and in general are never more than 5 since the convolution becomes computationally prohibitive.

1.3.1 Noise Filtering

The simplest method of filtering noise is to apply an averaging mask unconditionally at every pixel in an image. This approach has the disadvantage that in many instances the mask is applied where noise does not exist and hence image quality may be degraded as a result. Far better would be to first test to see if filtering is warranted at a given point before applying the mask. A simple method to achieve this (ref [1]) supposes that any given pixel in an image can be characterized by three components, an average component, a noise component and a non-noise or image component. These three components are characterized by three convolution masks applied to a 3×3 patch about the pixel. The definition of the first two masks is intuitively obvious while the third can be considered to be that which remains when the first two components have been removed. The averaging mask is defined as:

$$w_a = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



Figure 3: False colour image highlighting the vortex burst.

and the noise component mask is chosen such that it is orthogonal to the averaging mask.

$$\mathbf{w}_n = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Now if \mathbf{x} is the 3×3 'patch' around a pixel then the relative 'noise' component of the pixel is given by:

$$x_n = \frac{1}{|\mathbf{w}_n||\mathbf{x}|} \mathbf{w}_n * \mathbf{x}$$

and similarly the relative 'average' component is given by:

$$x_a = \frac{1}{|\mathbf{w}_a||\mathbf{x}|} \mathbf{w}_a * \mathbf{x}$$

From these results the relative image component of the pixel, x_i can be determined using the fact that,

$$x_n^2 + x_i^2 + x_a^2 = 1$$

x_n and x_i may now be compared to give an indication of the relative 'noisiness' of the pixel in question. If the pixel is deemed to be noisy its value can be replaced by the convolution,

$$f_r = \frac{1}{|\mathbf{w}_r|^2} \mathbf{w}_r * \mathbf{x}$$

where \mathbf{w}_r is a restoring mask given by:

$$\mathbf{w}_r = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

A useful measure of noisiness is given by:

$$\begin{aligned} \mu &= \frac{x_n^2}{x_n^2 + x_i^2} \\ &= \frac{x_n^2}{1 - x_a^2} \end{aligned}$$

Since $\mu \in [0, 1]$, an absolute threshold can be set between these two limits defining an acceptable level of noisiness. If μ is greater than this threshold, then the pixel is replaced by f_r , otherwise no change is made. Setting a threshold of 0 means that every pixel is convolved with the restoring mask, while a value of 1 implies that no change in the image takes place.

Up to this point the image intensity has been assumed to be a scalar function of array position. This is a valid assumption for gray scale images but not for colour images where each image pixel is defined by a three component vector. In the case of colour images the technique is simply applied to each of the three components in turn.

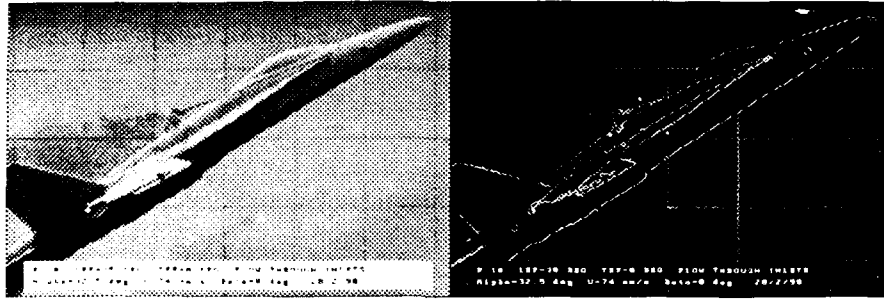


Figure 4: These two pictures illustrate the effect of edge enhancement using a 2×2 convolution mask

1.3.2 Edge Enhancing

In some instances only the boundaries between various regions within an image need to be highlighted, while all other colour or intensity information is suppressed. The simplest way to achieve this is to apply 2×2 "differencing" convolution masks to every pixel in the image. The masks commonly used are,

$$d_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad d_2 = \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix} \quad d_3 = \begin{bmatrix} 1 & 0 \\ -1 & 0 \end{bmatrix}$$

Any or all of these masks may be applied to each of the three colour components in an image and the results combined to give a measure of the edge characteristic at a given point in the image. Generally it is not desirable to maintain three separate edge measures for the three colour components. A single measure of colour difference is usually more meaningful than three separate measures for the three colour components, since colour images are not perceived as three separate images. An edge enhanced image may be output as a gray scale image with intensity proportional to the difference between neighbouring pixels. The image in Figure 4 was generated by first determining the maximum absolute difference between each pixel and its "forward" neighbours for each colour component. The three results are then summed together to give the final value on which the output is based.

A method similar to that described above for noise filtering can be applied to edge enhancing. (ref [2]) In this case two masks are used to characterize a pixel's edge component.

$$w_1 = \begin{bmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix} \quad w_2 = \begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix}$$

The additional work involved in using this technique however cannot be justified, since there is no significant improvement in output over the simpler, less computationally demanding methods using 2×2 convolution masks.

1.4 Spectral Techniques

Many useful techniques in image processing involve taking a spectral or Fourier transform of an image. Its applications revolve mainly around image restoration and noise filtering, although the method may be applied to contrast enhancement, edge enhancement and pattern recognition.

The form of the Fourier transform for a function of a single variable is well known.(ref [4]) In its complex form it is given by,

$$\mathcal{F}\{f(x)\} = F(u) = \int_{-\infty}^{\infty} f(x) \exp[-2\pi i u x] dx$$

The Fourier transform of a function of two variables is a logical extension of this definition and is given by,

$$\mathcal{F}\{f(x, y)\} = F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \exp[-2\pi i (ux + vy)] dx dy$$

The discrete two dimensional Fourier transform for a function defined on a rectangular domain can be inferred from the above definition.

$$\mathcal{F}\{f_{x,y}\} = F_{u,v} = \frac{1}{M \cdot N} \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} f_{x,y} \exp \left[-2\pi i \left(\frac{ux}{M} + \frac{vy}{N} \right) \right]$$

Here the function is assumed to be defined at $M \times N$ regular grid points separated by Δx and Δy in the x and y directions respectively. The discrete inverse transform can be shown to be given by:

$$\mathcal{F}^{-1}\{F_{u,v}\} = \sum_{v=0}^{N-1} \sum_{u=0}^{M-1} F_{u,v} \exp \left[2\pi i \left(\frac{ux}{M} + \frac{vy}{N} \right) \right]$$

It can be shown that $\mathcal{F}^{-1}\{F_{u,v}\} = f_{x,y}$ simply by substituting in the definition of $F_{u,v}$ and reversing the order of summation.

A significant feature of the two dimensional Fourier transform is that it is separable in the two variables.

$$\begin{aligned} F_{u,v} &= \frac{1}{N} \sum_{y=0}^{N-1} \left\{ \frac{1}{M} \sum_{x=0}^{M-1} f_{x,y} \exp \left[-2\pi i \frac{ux}{M} \right] \right\} \exp \left[-2\pi i \frac{vy}{N} \right] \\ &= \frac{1}{N} \sum_{y=0}^{N-1} F_{u,y} \exp \left[-2\pi i \frac{vy}{N} \right] \\ &= \mathcal{F}_y \{F_{u,y}\} \end{aligned}$$

Thus a two dimensional Fourier transform can be seen as two successive one dimensional transforms. The one dimensional transforms can be calculated using standard fast Fourier transform methods.

In general the function $F_{u,v}$ is complex. It should also be noted that $F_{u,v}$ is bi-periodic with $F_{u+M,v} = F_{u,v} = F_{u,v+N}$. If $f_{x,y}$ is real then it can be seen that $F_{-u,-v} = \overline{F_{u,v}}$. A similar rule applies to the inverse transform, that is if $F_{-u,-v} = \overline{F_{u,v}}$ everywhere then the inverse transform is real. Functions that define digital images are required to be real in physical space and their spectral transforms must therefore satisfy the condition of conjugate symmetry. Hence any modifications made to an image's spectral transform must be applied symmetrically.

1.4.1 Spectral Filtering

The most useful application of the spectral transform is that it can be used to filter or attenuate unwanted spectral components. For example spurious noise superimposed on an image generates high frequency components in the transform. Thus by filtering out these high frequency components, the noise in the resultant image can be removed. Note that the highest frequency in a transform image is given by $F_{M/2,N/2}$ and not $F_{M,N}$ since this is in fact equivalent to $F_{0,0}$. For this reason transforms are normally defined on the domain $u \in [-M/2 \dots M/2]$, $v \in [-N/2 \dots N/2]$. Low pass filtering has the undesirable side effect that the definition of edges in an image are also dependent on high frequency spectral components. Thus a balance must be struck between eliminating noise and degradation of image 'crispness'.

If it were required that the 'detail' of an image be highlighted, then a high pass filter can be applied to the spectral transform. Here low frequency components are attenuated with the result that edges, image structure and fine detail are given greater emphasis in the filtered image.

If an image is corrupted with some periodic or coherent noise, the image can possibly be restored using some special styles of filter which eliminate specific noise components in the spectral transform. This method is most effective in removing video scan lines or fringing effects created when digitizing an image from a video tape. The technique is also useful in suppressing interference fringes which tend to be generated when using laser sheet optics. Figure 5-12 show the effectiveness of spectral filtering. Figure 5 is a close up of the vortex burst flow visualization. The scan lines cause a significant loss of resolution of the image and consequently the image is useless for comparative purposes. Figure 7 shows the restored image which is a better representation of the original scene. Figure 9 shows an image taken from a video recording of a wind tunnel experiment in which smoke was introduced and illuminated using a laser sheet. The image is primarily corrupted as a result of non-uniform lighting produced by the use of an imperfect cylindrical lens in generating the sheet. These imperfections cause a series of parallel shadows to appear across the flow field. The image is further corrupted with video interference fringes. Figure 12 shows the image with both sources of interference removed.



Figure 5: A close-up of the vortex burst from figure 1. Note the presence of video scan lines in the image.

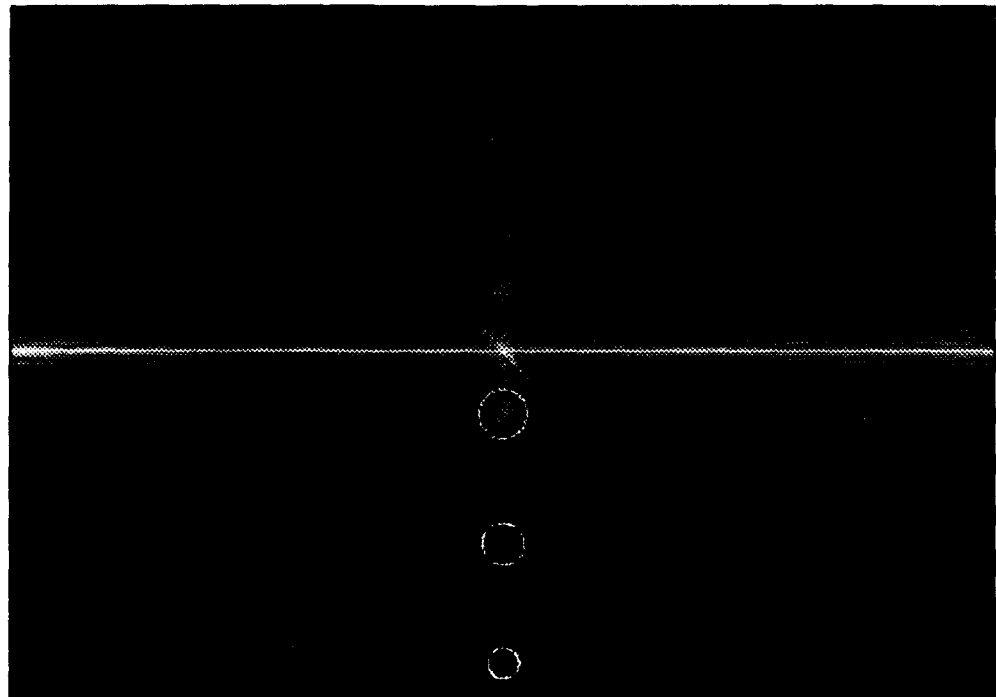


Figure 6: Spectral transform of figure 5. The bright circled regions are due to the scan lines in the original image. Note the bright regions appear symmetrically in the upper and lower halves of the transformed image.

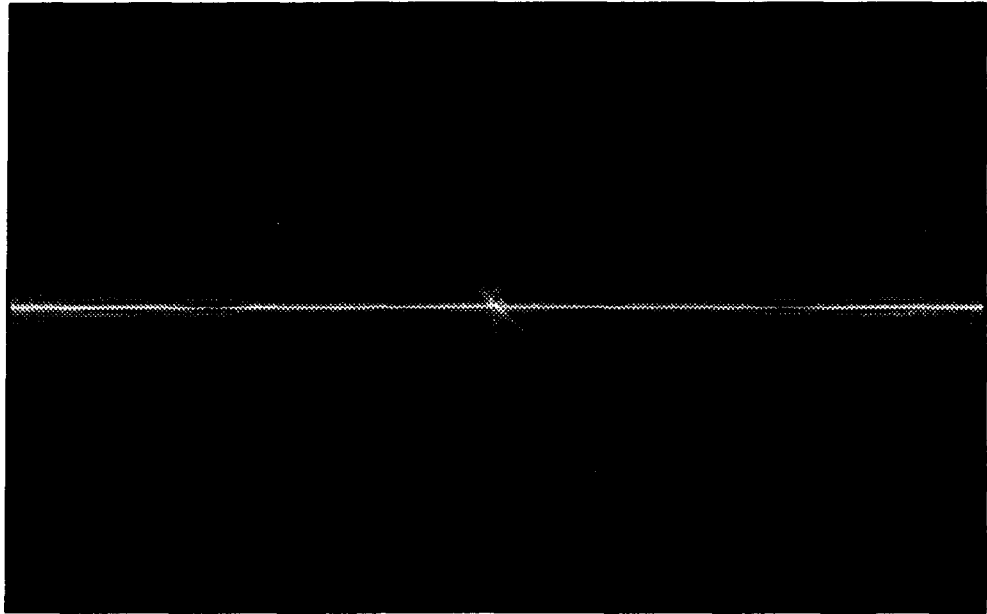


Figure 7: Spectral transform with scanline components filtered out



Figure 8: Restored image with scan lines removed



Figure 9: This image was taken from a video recording of a windtunnel experiment in which smoke was illuminated with a laser sheet. Note the parallel shadows caused by this illumination. The image was further corrupted with fringes when digitized from video tape.

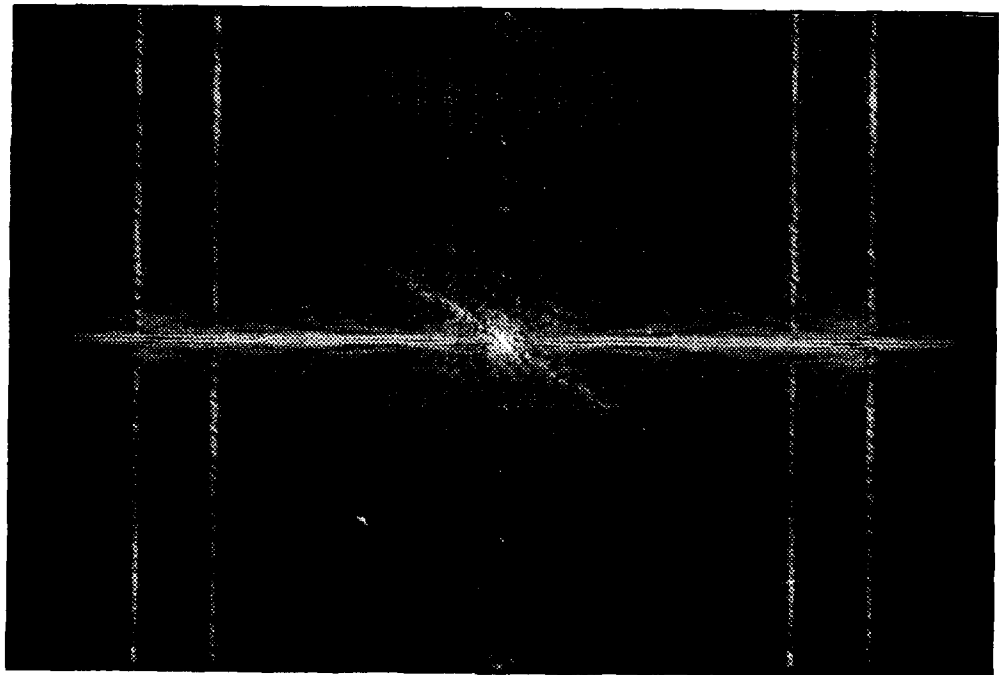


Figure 10: The spectral transform of the image in Figure 9 shows two flares emanating from the centre which are perpendicular to the shadows in the original image..

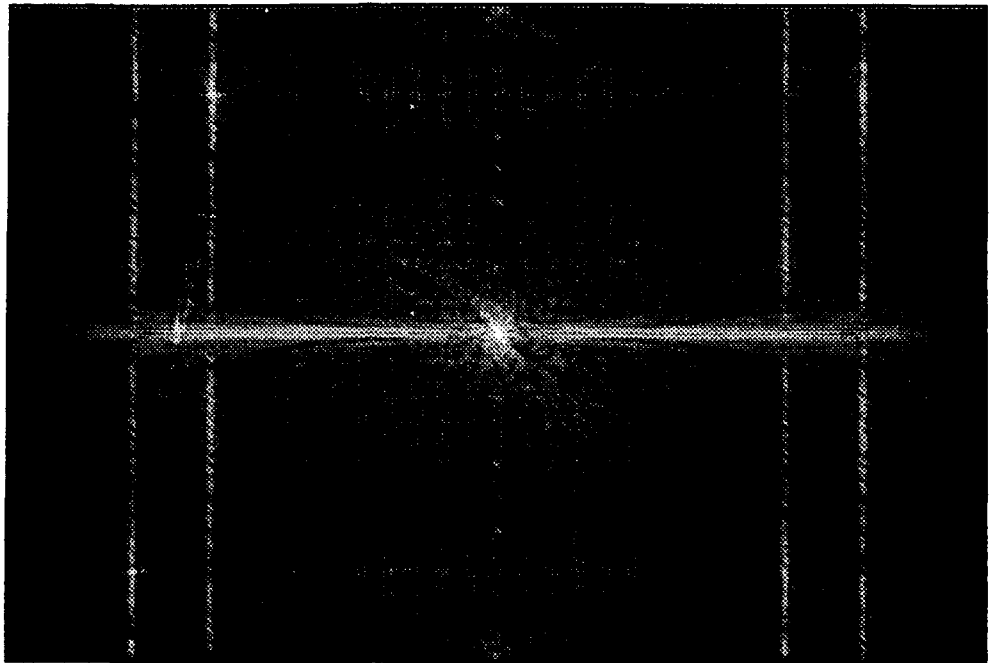


Figure 11: This image shows the spectral transform with the flares removed.



Figure 12: Restored image with shadows and video fringes removed.

1.4.2 Spectral Convolution

The convolution theorem of spectral transforms states that:

$$\begin{aligned}\mathcal{F}\{f * g\} &= FG \\ \Leftrightarrow f * g &= \mathcal{F}^{-1}\{FG\}\end{aligned}$$

Thus if a large convolution mask, 9×9 say, is to be applied to an image, then it may be more efficient to perform the convolution in spectral space. The process is then reduced to a single multiplication per pixel instead of 81 multiplications and summations per pixel. The overheads of course involve transforming and untransforming the image and thus the method is only beneficial if the convolution mask is large.

1.5 Edge Detection

Much work has been done on the problem of edge detection and image segmentation.(ref [3]) the problem is more difficult than one would first imagine. The heart of the problem is essentially how to define an edge in some meaningful way that a program might detect it. An edge essentially separates two distinct regions in an image, these regions may be characterized by different shades or different textures. The edge may not be clearly defined, the transition between one region and the next could be spread over a finite gap between them rather than a definite discontinuity. The problem is in fact impossible to automate in such a way that it will always agree with a human interpretation. The scope of the problem must be restricted somewhat before any simple solution can be devised. The method presented provides adequate edge detection between two regions bounded by a reasonable discontinuity in the distribution of gray scale intensity.

In general it is not possible to discern the location of an edge merely by examining one pixel's immediate neighbours. What might be interpreted as an edge in the context of a small group of pixels may be meaningless in the larger context. It is thus important to consider a reasonably sized neighbourhood when detecting an edge. However as the sample size is increased, the resolution of segment detection is correspondingly decreased. Hence a balance is required to achieve the desired results and match the scale of the image segment. A simple edge definition algorithm can be constructed in two phases. First determining the location of an edge by examining reasonably sized samples of the image and second defining the edge boundary within the context of the sample.

In order to automate the edge detection process, an absolute measure of an edge must be constructed. If a sample of pixels is taken in the vicinity of an edge, then the distribution of gray scale intensities in this sample may look something like the distribution in Figure 13b). The standard deviation of this distribution is about 2-3 R, where R is the range of intensities in the sample. (i.e. the difference between the maximum intensity in the sample and the minimum intensity) If a sample is taken where no edge exists the sample may look like the distribution in Figure 13a). Here

the standard deviation is about 4-5 R. If a sample is taken on a region which is not an edge but has smoothly varying gray scale such as might be seen on the surface of a cylinder the distribution may look like the distribution in Figure 13c). In this case the standard deviation is about 3-4 R. Thus the ratio of standard deviation to range gives a reasonable indication of the existence of an edge provided the sample is taken with roughly equal representation from the regions on either side. A problem arises however when the sample is taken at a point where three regions meet here the distribution is something like the distribution in Figure 13d). The standard deviation ratio for this distribution is the same as that of Figure 13c). Thus it would not be possible using this method to discern between these two cases.

The next step is to define the exact boundary line between one region and the next. The boundary is defined as a continuous line that runs *between* pixels in an image. Once an edge has been detected within a small region the two extremum pixels in that sample, with the maximum and minimum intensities, are assumed to be on either side of the boundary. A region is grown from each of these pixels by adding a neighbouring pixel with the minimum difference in intensity. The region is stopped when the next pixel is closer in intensity to the opposite extremum pixel than its neighbouring pixel which is part of the grown region. Once the two separate regions have been defined within the sample region they are checked using the ratio of standard deviation to range to ensure that they are not further subdivisible. If either of the defined regions is in fact two regions then a boundary is formed between these subdivisions. This process continues until all edges in the sampled region have been defined. If an edge is found to continue beyond the sampled region, the neighbouring region is then divided until a continuous boundary around the selected image segment is found. When this is achieved and no further edges exist within the boundary, a region can be grown which from a pixel known to exist within the required image segment. The region will consist of all pixels that can be reached by an unbroken path to the seed pixel without crossing a defined edge.

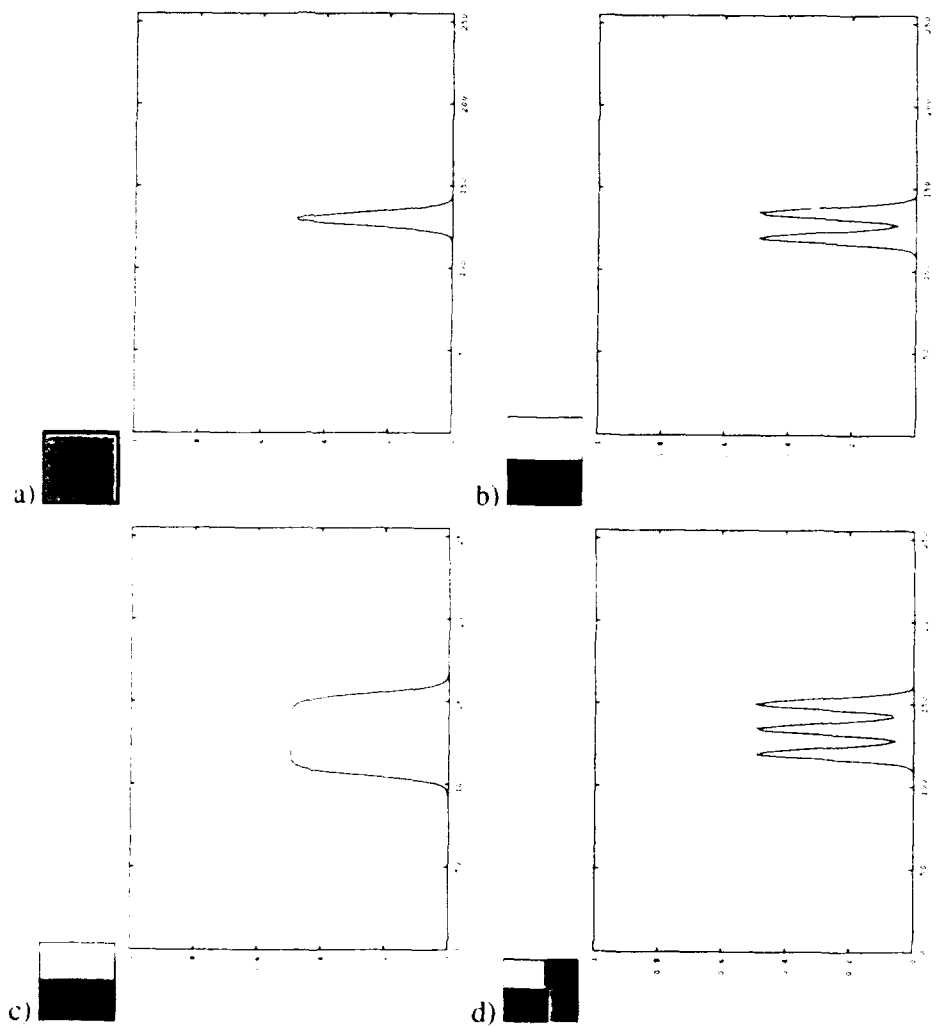


Figure 13: Intensity distributions of small sampled regions of an image.

2. The Pixedit Program

Pixedit is a program used for the editing and creation of image files for display on Silicon Graphics IRIS workstations. The original version of the program was written at NASA Ames by D. Choi. A new version has been written to make use of the features of the IRIS 4D series of workstations. Consequently the new version is faster and easier to use than the old version, version 2.3. The new version includes new routines which allow processing and image restoration of gray scale images and, in the case of some routines, colour images.

The new version accepts three types of image file format, the standard Silicon Graphics image file format characterized by the extension '.rgb' or '.gray', an 'RImage' type format utilized by the previous version of *Pixedit* and a format utilized by the PC program "ImageAction Version 2.0" characterized by the '.img' extension. This last format was included so that image files generated from PC based video frame-grabbing systems could be read directly by the new version. Output files from the program are of the standard image file format compatible with most IRIS 4D graphics packages that utilize raster images. Note that utilities exist to convert this format to most other useful image file formats that may be required.(i.e. targa giff tiff etc.)

The new features of *Pixedit* primarily for the processing of gray scale images. These processes include edge enhancement and edge detection, a noise filter, Fourier transformations and filtering, gray scale histogram manipulation and false colour enhancing. Edge enhancement and noise filtering can be performed on colour images as well as on gray scale images.

This section gives a brief description of the old version of *Pixedit* as well as more comprehensive description of how to operate the new features of version 3.0.

2.1 The Main Menu

The program is controlled using a mouse operated menu system to selected the various picture editing functions. The main menu is selected by clicking the left mouse button anywhere on the background, *not* within the bounds of an image. The options in this menu are:

Read Image File Read a new image file from disk. The file should be one of the three formats specified above namely:

- Image Format (.rgb, .gray, .bw, etc)
- RImage Format (.RGB, .GRAY .BW, etc)
- PC Format (.img)

Snap New Image This allows all or part of the current screen display to be copied or 'snapped' into a new image.

Background Colour Specifies a new background colour the red, green and blue sliders are used to set the three colour components of the new background colour.

Toggle Grid Switches a square grid on or off. The grid fills the screen and is used to help align separate images on the screen.

Type in String Produces an image consisting of a typed string written in the system font. This 'string' image can be manipulated just like any other image.

Save Screen Saves the entire screen as an image file in the standard 'Image' format.

Exit Pixedit Exit from the Pixedit program.

2.2 The Image Menu

There is a second menu used to select features which operate specifically on individual images. This *image menu* is invoked by clicking, with the left mouse button, within the bounds of any given image. The selectable functions in this menu are:

Pop Makes the selected image the top image. The image is redrawn and any images behind it become obscured.

Push Makes the selected image the bottom image. Any part of this image that obscures another is erased.

Move The selected image can be moved to a new position on the screen.

Duplicate A copy of the selected image is drawn on the screen. It should be noted that only one copy of the image data is stored in memory.

Rotate The selected image can be rotated about any axis with or without perspective.

Resize/Reshape Rescale the selected image in the horizontal and/or vertical direction.

Paint Paint over parts of the selected image in a selected colour.

Modify Colour Add colour to all or part of a selected gray scale image or change a particular colour of a colour image.

Gray Scale Converts the selected colour image to a gray scale image by assigning each colour component, red, green and blue, a gray scale value.

Transparency Combine the selected image with the images drawn behind it by making all or part of the image transparent or by interleaving the images together.

Filters Used to enhance edges, filter noise, lighten or darken the selected image.

Fourier Transform Generates a spectral transform of the selected image. This transform can be filtered using various styles of filter apertures and then inverse transformed thus removing unwanted spectral properties of an image.

Save Image Write the selected image to standard image file format on disk.

Delete Image Remove the selected image from the screen and delete it from memory.

The use and function of some of these features is self evident and require no further explanation. Operations which require some explanation are detailed in the following sections.

2.3 The Image In Memory

A colour raster image is stored as a rectangular array of colour vectors, each colour vector being composed of red, green and blue components. These components consist of numbers between 0 to 255. Thus the memory requirement for each colour component of a given pixel is 8 bits or 1 byte. The IRIS 4D provides an extra 'colour' component which, loosely speaking, is used to specify the opacity of a pixel. Although this fourth component is generally superfluous for simple image displays, space must be provided for it. Thus 4 bytes of memory are required for each pixel in the image.

The extra byte in each pixel, referred to as the alpha byte, is utilized by the Pixedit program to indicate whether a pixel is a gray pixel or a coloured pixel. If the pixel is coloured, 0 is stored in the alpha byte, otherwise the gray scale value of the pixel is stored. The distinction between gray and colour pixels is necessary since a gray pixel is assigned it's colour from a look-up table (LUT) for false colouring purposes. The appropriate look-up table entry is stored in the red, green and blue bytes of the pixel. The default colour for a gray pixel has red, green and blue components equal to that of the gray scale or alpha value. Thus the colour appearance of a given 'gray' pixel may be changed but its gray scale value remains fixed.

An image may be one of two types, an RGB image or a GRAY image. GRAY images contain no colour pixels while an RGB image may have both 'gray' and 'colour' pixels. When a gray image is saved to disk it can be written in one of two formats:

- As a 'gray' image where only the gray scale values and the colour look-up table are saved.
- As an 'rgb' image where the current appearance of the image is saved and the gray scale values are ignored.

RGB images can only be written in the second of these formats.

2.4 The *Paint* Function

This feature allows the selected image to be painted over using a mouse controlled paint brush. The size and shape of the brush and the colour of the paint are selected using the slider control. The first two sliders, coloured magenta and yellow, define the height and width of the brush while the next three sliders define the red, green, and blue colour components of the painting colour.

The sliders may be set manually by clicking with the left mouse button in the sliders bounding rectangle. The painting colour can be selected by clicking the left mouse button on a pixel with the desired colour. Thus a colour can be chosen to match a colour present in the image under consideration or in any other image.

Painting a gray scale image with any colour (even a shade of gray) causes that image to be regarded as a colour image. The image may be converted back to gray scale image by use of the 'Gray Scale' function found in the Image Menu.

2.5 The *Modify Colour* Function

This feature is used to change the colour appearance of GRAY or RGB images. There are two methods of colour modification corresponding to the two different styles of pixels, gray and colour. Gray pixels are modified by use of a look-up table, while colour pixels are modified by specifying a target colour, and then specifying the colour to which that target colour is to be changed.

The entire image can be modified using the 'do all' button on the slider or areas can be modified selectively using a paint brush similar to that used for the *Paint* function. If only selected areas of a gray image are modified then the image becomes a colour image. The *Inverse do all* button will change all colour pixels except the specified target colour pixels to the new colour. The target colour is set by clicking with the left mouse button on a pixel of the required colour. The new colour is set using the red, green and blue sliders or by clicking with the middle mouse button on a pixel of the required colour.

The look-up table (LUT) for gray pixels is set using the tool on the right of the sliders. The two shaded rectangles represent two colour look-up tables, the colours correspond to gray scale values 0 (default black) at the bottom to 255 (default white) at the top. The rectangle on the left is the image's LUT while the one on the right is the current user defined LUT. If the *Do all* button is executed the user defined LUT replaces the image's LUT. If the selected image happens to be a GRAY type image, then the image's histogram is displayed in red to the left of the image LUT. This histogram shows the relative occurrence of each gray scale value in the image.

There are three menus associated with this tool, the first menu is accessed by clicking the right mouse button on the left LUT rectangle. This menu contains a number of predefined LUT's which can be selected and made the current LUT. The current user defined LUT can be saved with the *Save* option and recalled at a later time with the *User Defined* option from this menu. The second menu, the user LUT menu,

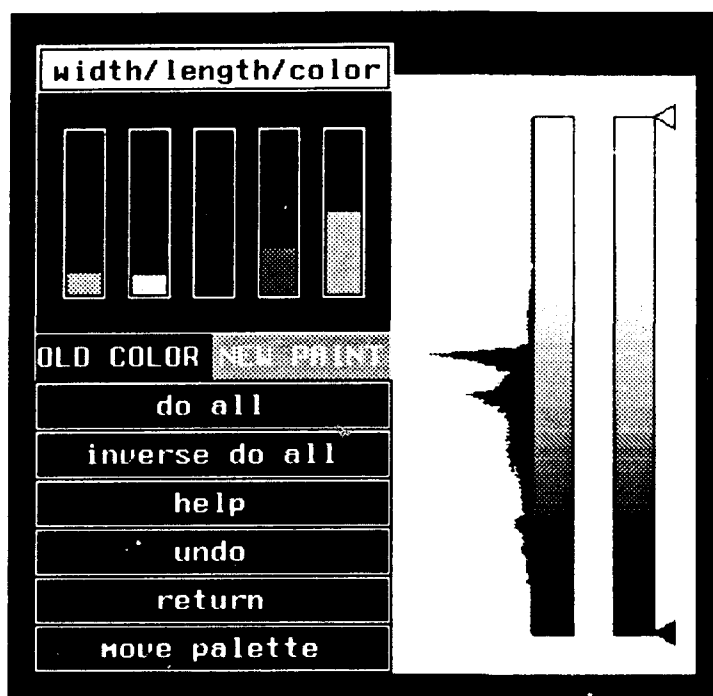


Figure 14: The *Modify Colour* tool.

is initiated by clicking with the right mouse button on the right LUT rectangle. Options from this menu effect the user LUT globally. The third menu is the marker menu and is associated with each individual marker. It is initiated by clicking the right mouse on the required marker and will effect only the marker selected.

The markers are used to define the colour at the marker location, they can also be used to position a contour band in the LUT. The marker position can be changed by clicking on them with the left mouse button. Colours are linearly interpolated between separate markers, a marker's colour is changed using the marker's menu. Each marker has two colours, a top and a bottom colour, if these colours are the same then a smooth variation of colour is seen but if they are not the same then a colour discontinuity in the LUT results. When either of the colour changing options is chosen the marker is highlighted, the new colour of the marker can now be set using the colour slider or by clicking on the required colour somewhere on the screen with the left mouse button. To unselect a marker click on it with the left mouse button. When a change of either of the marker's colours is initiated, the colour to be changed is set equal to the marker's other colour. If the two colours are the same the colour to be changed is set equal to that of the marker above or below it, which ever is appropriate. There are two special markers at the top and bottom of the LUT rectangle, these markers are fixed and have only one colour for obvious reasons.

Each marker can have a 'contour' associated with it. When a marker's contour width is non-zero a number of gray scales in the vicinity of the marker are mapped to the

contour colour. The contour width can be set for an individual marker by selecting the required contour width on the marker's menu or the contour width can be set globally for all markers by selecting the contour width from the user LUT menu. The contour colour is set for all markers from the user LUT menu.

2.6 The Filter Function

The processes available under the Filter function are equally applicable to both 'colour' and 'gray' images. The processes include:

- Edge Enhancement
- Noise Filtering
- Lighten Image
- Darken Image

Edge enhancement and noise filtering are performed using a convolution process applied successively to a small patch (2x2 or 3x3 pixels) in the neighbourhood of each pixel in the image. The Noise Filtering option uses two successive convolutions the first gives a measure of the noise characteristic of a given pixel and if this measure is above a user defined threshold then the second averaging convolution is applied. The threshold for each colour component is set using the red, green and blue sliders. If the sliders are set to zero then the filter has no effect on the image while if they are set to the maximum then all pixels are filtered regardless.

2.7 The Fourier Transform Function

The Fourier Transform function performs a discrete spectral transform on gray scale images and displays a representation of this transform. The transformation equation is given by,

$$F(u, v) = \frac{1}{MN} \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} f(x, y) \exp \left[-2\pi i \left(\frac{ux}{M} + \frac{vy}{N} \right) \right]$$

where $f(x, y)$ is an integer valued function giving the gray scale values of a gray image and $F(u, v)$ is the resulting transform function. It should be noted that whilst $f(x, y)$ is a real valued function, $F(u, v)$ is complex thus the displayed representation of the transform shows only magnitude of the transform function. The transform can be displayed in one of two ways:

1. Normal display where the display intensity is proportional to the magnitude of the transform.

2. Log display where the displayed intensity is proportional to $\log(1 + |F(u, v)|)$. This is useful since the range of values in the transform spans several orders of magnitude. This display enables the smaller values to be visible in the transform display.

To select between these two display modes click the right mouse button anywhere to bring up the Transform Menu. Note that the current display mode is disabled and cannot be selected. The displayed image will always appear symmetric since $F(-u, -v) = \overline{F(u, v)}$ when $f(x, y)$ is real and thus $|F(-u, -v)| = |F(u, v)|$.

By entering the *Filters* submenu a number of filters can be laid over the transform. These filters effectively set the value of the transform to zero over the filtered region. Such filters are useful in suppressing high frequency noise, removing video scanlines or fringing effects or any periodic interference in an image. By filtering out the low frequency region, the edges of the resultant image can be highlighted.

Once filtering has been completed the filtered image is restored by selecting the *Untransform* option from the transform menu. This will also take some time to compute. The inverse transform is given by,

$$f(x, y) = \sum_{v=0}^{N-1} \sum_{u=0}^{M-1} F(u, v) \exp \left[2\pi i \left(\frac{ux}{M} + \frac{vy}{N} \right) \right]$$

To exit from the transform mode select *Quit* from the transform menu.

3. Conclusion

Pixedit version 3.0 can now be used to process images obtained from the flight mechanics water tunnel or any image in the standard IRIS image type format. The program provides an environment where images can be interactively modified or combined with other images for presentation or informative purposes. *Pixedit* provides a means by which the quality of digital images can be significantly improved. Both in their aesthetic appearance and in their capacity to convey significant information in a short space of time.

It is envisaged that further modifications to the program may permit indivisible segments within an image to be selected and once defined an image segment may be processed separately from the rest of the image. A useful feature would be that given the definition of an image segment, say the marker spots on the F-18 fuselage, the program may be able to search the image for that segment in a different image. This would permit the segment to be traced through a series of frames from a video recording, thus providing quantitative experimental data.

References

- [1] R. C. Gonzalez & P. Wintz, *Digital Image Processing*, 2nd ed. U.S.A., Addison-Wesley, 1987.
- [2] T. Pavlidis, *Algorithms for Graphics and Image Processing*, Rockville, U.S.A., Computer Science Press, 1982.
- [3] T. Pavlidis, *Structural Pattern Recognition*, Eaglewood Cliffs, New Jersey: Prentice-Hall, 1975.
- [4] L.D. Maclaren & S.D. Hill, *A Fortran Program for Spectral Analysis Using The Fast Fourier Transform*, ARL-FLIGHT-MECH-TM-432, AR-006-583, 1991.

DISTRIBUTION

AUSTRALIA

Department of Defence

Defence Central

Chief Defence Scientist)
AS, Science Corporate Management)shared copy
FAS Science Policy)
Director, Departmental Publications
Counsellor, Defence Science, London (Doc Data sheet only)
Counsellor, Defence Science, Washington (Doc Data sheet only)
Scientific Adviser, Defence Central
OIC TRS, Defence Central Library
Document Exchange Centre, DSTIC (8 copies)
Defence Intelligence Organisation
Librarian H Block, Victoria Barracks, Melb (Doc Data sheet only)

Aeronautical Research Laboratory

Director
Library
Chief Flight Mechanics and Propulsion Division
Branch File - Flight Mechanics
Branch Head - Flight Mechanics
Author: Michael Lee
B. Fairlie
J. Lopez
J. Bulbeck
S. Henbest
D. Thompson

Defence Science & Technology Organisation - Salisbury Library

Navy Office

Navy Scientific Adviser (3 copies Doc Data sheet only)

Army Office

Scientific Adviser - Army (Doc Data sheet only)

Air Force Office

Air Force Scientific Adviser (Doc Data sheet only)
Aircraft Research and Development Unit
Scientific Flight Group
Library

SPARES (5 COPIES)

TOTAL (32 COPIES)

DOCUMENT CONTROL DATAPAGE CLASSIFICATION
UNCLASSIFIED

PRIVACY MARKING

1a. AR NUMBER AR-006-656	1b. ESTABLISHMENT NUMBER ARL-FM-TM-452	2. DOCUMENT DATE DECEMBER 1991	3. TASK NUMBER DST 91/041	
4. TITLE DIGITAL ENHANCEMENT AND RESTORATION OF IMAGES DERIVED FROM FLOW VISUALIZATION EXPERIMENTS		5. SECURITY CLASSIFICATION (PLACE APPROPRIATE CLASSIFICATION IN BOX(S) (E. SECRET (S), CONF. (C) RESTRICTED (R), LIMITED (L) UNCLASSIFIED (U)). <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px; text-align: center;">U DOCUMENT</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">U TITLE</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">U ABSTRACT</div> </div>	6. NO. PAGES 28	7. NO. REFS. 4
8. AUTHOR(S) M.F. LEE		9. DOWNGRADING/DELIMITING INSTRUCTIONS Not applicable		
10. CORPORATE AUTHOR AND ADDRESS AERONAUTICAL RESEARCH LABORATORY 506 LORIMER STREET FISHERMENS BEND VIC 3207		11. OFFICE/POSITION RESPONSIBLE FOR: SPONSOR _____ DSTO SECURITY _____ - DOWNGRADING _____ - APPROVAL _____ CFPD		
12. SECONDARY DISTRIBUTION (OF THIS DOCUMENT) Approved for public release OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DSTIC, ADMINISTRATIVE SERVICES BRANCH, DEPARTMENT OF DEFENCE, ANZAC PARK WEST OFFICES, ACT 2601				
13a. THIS DOCUMENT MAY BE ANNOUNCED IN CATALOGUES AND AWARENESS SERVICES AVAILABLE TO ... No limitations				
13b. CITATION FOR OTHER PURPOSES (IE. CASUAL ANNOUNCEMENT) MAY BE		<input checked="" type="checkbox"/> UNRESTRICTED OR	<input type="checkbox"/> AS FOR 13a.	
14. DESCRIPTORS Flow visualisation Digital image processing Image enhancement			15. DISCAT SUBJECT CATEGORIES 2004 120902	
16. ABSTRACT <i>The Pixedit program has been rewritten to enable image processing of raster image files residing on the IRIS 4D workstations. Such techniques as spectral filtering, edge detection and enhancement and false colouring techniques have been incorporated into the new version of the program. These techniques are usefully applied in enhancing the quality of photographic images obtained from flow visualization experiments. This report describes the techniques used, giving examples of their application to flow visualization and provides instructions for operating the revised Pixedit program</i>				

PAGE CLASSIFICATION
UNCLASSIFIED

PRIVACY MARKING

THIS PAGE IS TO BE USED TO RECORD INFORMATION WHICH IS REQUIRED BY THE ESTABLISHMENT FOR ITS OWN USE BUT WHICH WILL NOT BE ADDED TO THE DISTS DATA UNLESS SPECIFICALLY REQUESTED.

16. ABSTRACT (CONT):

17. IMPRINT

AERONAUTICAL RESEARCH LABORATORY, MELBOURNE

18. DOCUMENT SERIES AND NUMBER

Flight Mechanics Technical
Memorandum 452

19. COST CODE

53 509F

20. TYPE OF REPORT AND PERIOD COVERED

21. COMPUTER PROGRAMS USED

22. ESTABLISHMENT FILE REF (S)

23. ADDITIONAL INFORMATION (AS REQUIRED)